$ 50 fc

PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

09/842,312

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	Patent#: 7047399
Filing Date	Issued: May 16, 2006
First Named Inventor	Andrew C. Sturges
Art Unit	2183
Examiner Name	Richard L. Ellis
Total Number of Pages in This Submission: 9	Attorney Docket Number: S1022.80655US00

ENCLOSURES (*Check all that apply*)

- [x] Check for $100.00
- [ ] Fee Attached
- [ ] Amendment/Reply
- [ ] After Final
- [ ] Affidavits/declaration(s)
- [x] Request for Certificate of Correction
- [x] Certificate of Correction
- [x] Cols. 3, 4, 6 and 17 of 7,047,399
- [ ] Certified Copy of Priority Document(s)
- [ ] Reply to Missing Parts/ Incomplete Application
- [ ] Reply to Missing Parts under 37 CFR 1.52 or 1.53
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
- [ ] Landscape Table on CD
- [ ] After Allowance Communication to TC
- [ ] Appeal Communication to Board of Appeals and Interferences
- [ ] Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- [ ] Proprietary Information
- [ ] Status Letter
- [ ] Other Enclosure(s) (please Identify below):

Return Post Card

Remarks

Certificate
MAY 31 2006
of Correction

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	WOLF, GREENFIELD & SACKS, P.C.		
Signature			
Printed name	Daniel P. McLoughlin		
Date	May 24, 2006	Reg. No.	46,066

Certificate of Mailing Under 37 CFR 1.8(a)

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the U.S. Postal Service on the date shown below with sufficient postage as First Class Mail, in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Dated: May 24, 2006

Signature: _____ (June M. Watson)

MAY 31 2006

PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	Patent#: 7047399
Filing Date	Issued: May 16, 2006
First Named Inventor	Andrew C. Sturges
Art Unit	2183
Examiner Name	Richard L. Ellis
Total Number of Pages in This Submission: 9	Attorney Docket Number: S1022.80655US00

ENCLOSURES (*Check all that apply*)

- [x] Check for $100.00
- [ ] Fee Attached
- [ ] Amendment/Reply
- [ ] After Final
- [ ] Affidavits/declaration(s)
- [x] Request for Certificate of Correction
- [x] Certificate of Correction
- [x] Cols. 3, 4, 6 and 17 of 7,047,399
- [ ] Certified Copy of Priority Document(s)
- [ ] Reply to Missing Parts/ Incomplete Application
- [ ] Reply to Missing Parts under 37 CFR 1.52 or 1.53
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
- [ ] Landscape Table on CD
- [ ] After Allowance Communication to TC
- [ ] Appeal Communication to Board of Appeals and Interferences
- [ ] Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- [ ] Proprietary Information
- [ ] Status Letter
- [ ] Other Enclosure(s) (please Identify below):

Return Post Card

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	WOLF, GREENFIELD & SACKS, P.C.		
Signature			
Printed name	Daniel P. McLoughlin		
Date	May 24, 2006	Reg. No.	46,066

Certificate of Mailing Under 37 CFR 1.8(a)

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the U.S. Postal Service on the date shown below with sufficient postage as First Class Mail, in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Dated:May 24, 2006 Signature: _____ (June M. Watson)

Docket No.: S1022.80655US00
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Andrew C. Sturges and Nathan M. Sidwell
Serial No.:	09/842312
Confirmation No.:	6679
Filed:	April 25, 2001
Patent No.:	7047399
For:	COMPUTER SYSTEM AND METHOD FOR FETCHING, DECODING AND EXECUTING INSTRUCTIONS
Examiner:	Richard L. Ellis
Art Unit:	2183

Certificate of Mailing Under 37 CFR 1.8(a)
I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the U.S. Postal Service on the date shown below with sufficient postage as First Class Mail, in an envelope addressed to: Attention: Certificate of Correction Branch, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Dated: May 24, 2006

June M. Watson

REQUEST FOR CERTIFICATE OF CORRECTION PURSUANT TO 37 CFR 1.323

Attention: Certificate of Correction Branch
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Upon reviewing the above-identified patent, Patentees note typographical errors which should be corrected.

In the Specification:

Column 3, line 28, currently reads:

"provided a computer system for fetching, decoding **end"** (emphasis added)

Column 3, line 28, should read as shown below:

--provided a computer system for fetching, decoding **and--** (emphasis added)

The word "end" erroneously appears in column 3, line 28 of issued U.S. Patent No. 7,047,399. This should be changed to "and."

Column 4, line 32, currently reads:

"selected **cone** of said instruction fetchers to said execution" (emphasis added)

Column 4, line 32 should read as shown below:

--selected **one** of said instruction fetchers to said execution-- (emphasis added)

The word "cone" erroneously appears in column 4, at line 32 in issued U.S. Patent No. 7,047,399. This should be changed to "and."

Column 4, line 44, currently reads:

"can be done in a **umber** of ways. For example, a further" (emphasis added)

Column 4, line 44 should read as shown below:

--can be done in a **number** of ways. For example, a further-- (emphasis added)

The word "umber" erroneously appears in column 4, at line 44 in issued U.S. Patent No. 7,047,399. This should be changed to "number."

Column 4, line 49, currently reads:

"register can **ten** be compared with an instruction pointer" (emphasis added)

Column 4, line 49 should read as shown below:

--register can **then** be compared with an instruction pointer-- (emphasis added)

The word "ten" erroneously appears in column 4, at line 49 in issued U.S. Patent No. 7,047,399. This should be changed to "then."

Column 6, line 8, currently reads:

"commencing from target location. The rejected state is one"

Column 6, line 8 should read as shown below:

--commencing from **a** target location. The rejected state is one-- (emphasis added)

The word "a" was erroneously omitted from column 6, at line 8 of issued U.S. Patent No. 7,047,399. The word "a" should be inserted between "from" and "target."

Column 17, line 55, currently reads:

"kernel signal is asserted at the same time as the start **signals**" (emphasis added)

Column 17, line 55 should read as shown below:

-- kernel signal is asserted at the same time as the start **signal**-- (emphasis added)

The word "signals" erroneously appears in column 4, at line 49 in issued U.S. Patent No. 7,047,399. This should be changed to "signal."

The errors were found in the application as filed by applicant. Our check in the amount of $100.00 covering the fee set forth in 37 CFR 1.20(a) is enclosed.

The errors now sought to be corrected are inadvertent typographical errors the correction of which does not involve new matter or require reexamination.

Transmitted herewith is a proposed Certificate of Correction effecting such amendment. Patentee respectfully solicits the granting of the requested Certificate of Correction.

The Director is hereby authorized to charge any deficiency in the fees filed, asserted to be filed or which should have been filed herewith (or with any paper hereafter filed in this application by this firm) to our Deposit Account No. 23/2825, under Docket No. S1022.80655US00. A duplicate copy of this paper is enclosed.

Dated: May 24, 2006

Respectfully submitted,

By: _____

Daniel P. McLoughlin
Registration No.: 46,066
WOLF, GREENFIELD & SACKS, P.C.
Federal Reserve Plaza
600 Atlantic Avenue
Boston, Massachusetts 02210-2206
(617) 646-8000

instruction to effect a branch into two different parts. The set branch instruction indicates the target location for the branch and can be placed as near the beginning of the string of instructions as possible.

Actual implementation of the branch is carried out later in response to a split bit located. in a later instruction.

The provision of the target location for the branch with the set branch instruction provides an early indication of the fact that a memory access is going to be made (or is likely to be made) and provides the memory address (the target location) for that access. When the split bit causes the branch to be taken, and the time comes therefore to access that memory address, the system has had a chance to set up for the access, for example by bringing the necessary data into a local cache.

One problem associated with the system of EP-A-355069 is that the target location from which new instructions are fetched is reset after a split bit signal has been executed. This means that there cannot be multiple branches using the target location set up by a single set branch instruction. It is advantageous to allow for this situation and it is one object of the present invention to provide an improved system for implementing branches allowing for this.

SUMMARY OF THE INVENTION

According to one aspect of the present invention, there is provided a computer system for fetching, decoding end executing instructions comprising storage circuitry for holding a plurality of instructions at respective storage locations, said plurality of instructions being arranged in instruction strings, each string comprising a first instruction and a set of subsequent instructions, instruction fetch circuitry for fetching a sequence of instructions from said storage circuitry and including an indicator for providing an indication of a next address at which a next fetch operation is to be effected, execution circuitry for executing fetched instructions, wherein at least some of said instruction strings each includes a set branch instruction (SET) which provides an indication of a target location from which a subsequent instruction may be fetched, the subsequent instruction being from a different instruction string, and wherein said instruction fetch circuitry is operated responsive to execution of a said set branch instruction (SET) to fetch in parallel subsequent instructions from said string containing said set branch instruction and new instructions from said different instruction string commencing from said target location while said subsequent instructions continue to be executed. The computer system further comprises a target store for holding the indication of said target location, said indication being loaded into said store on execution of said set branch instruction (SET) and being held in said store as a valid indication until execution of a subsequent set branch instruction and select circuitry responsive to generation of an effect branch (DO) signal indicative that further instructions to be executed are said new instructions, to cause said execution circuitry to execute said new instructions and to cause said instruction fetch circuitry to fetch again new instructions commencing from said target location.

The invention also provides a method of operating a computer to fetch decode and execute instructions which computer has storage circuitry holding a plurality of instructions at respective storage locations, said plurality of instructions being arranged in instruction strings, each string comprising a first instruction and a set of subsequent instructions. The method comprises fetching a sequence of instructions from said storage circuitry and providing an indication of a next address at which a next fetch operation is to be effected, decoding said instructions, and executing each instruction in turn, wherein at least some of said instruction strings each include a set branch instruction (SET) which provides an indication of a target location from which a subsequent instruction may be fetched, the subsequent instruction being from a different instruction string. The method further comprises, on execution of said set branch instruction, holding the indication of said target location in a target store as a valid indication until execution of a subsequent set branch instruction, fetching in parallel subsequent instructions from the string containing said branch instruction and new instructions from said different instruction string commencing from said target location, continuing to execute said subsequent instructions until an effect branch signal is generated which indicates that further instructions to be executed are said new instructions, and responding to said effect branch signal by commencing execution of said new instructions and fetching again new instructions commencing from said target location.

In one embodiment, said instruction fetch circuitry comprises two instruction buffers, a first buffer for holding subsequent instructions connected to said execution circuitry, and a second buffer for holding new instructions wherein the contents of said second buffer are copied into said first buffer responsive to generation of said effect branch (DO) signal.

In the described embodiment said instruction fetch circuitry includes two instruction fetchers for fetching respectively said subsequent instructions and said new instructions and wherein said select circuitry is operable to connect a selected cone of said instruction fetchers to said execution circuitry.

In the simplest case, the target store can hold the memory address of the target location. To allow kernel entry, the set branch instruction can identify the target location using an implicit value which addresses a special register holding the memory address of the new instructions.

To allow descriptor branches to be executed, the target store can hold a pointer to a memory location which contains the memory address of the target location.

The effect branch signal is generated when the branch point, at which the branch is to be taken, is identified. This can be done in a umber of ways. For example, a further instruction can be located in the string of instructions being executed prior to the branch point in which case said further instruction will identify the branch point which will be held in a branch point register. The contents of the branch point register can ten be compared with an instruction pointer register holding an indication of the address from which a next instruction would normally be fetched and when the two are equal the effect branch signal is generated. Alternative methods for identifying the branch point are also discussed herein.

The provision of a further instruction which identifies the branch point but which is located before the branch point reduces the number of unwanted instructions which will be fetched before the branch is taken.

As a still further alternative, the set branch instruction itself can identify the branch point which is stored in the branch point register, thereby obviating the need for a further instruction.

However, in a particularly preferred embodiment, the branch point is identified by a further, dedicated instruction, different to the set branch instruction, which is located at the branch point in the string of instructions being executed. To allow for additional branches to be effected, this effect

branch instruction can itself define the condition to be satisfied so that a branch is only taken if the condition is satisfied and is not taken if the condition not satisfied.

This provides a further technical advantage over the system of EP-A-355069 discussed above. In that system, the set branch instruction itself must indicate whether or not the branch is conditional or not and cause various different condition detectors to be in a ready state, ready to sense a condition. The condition itself is defined in an instruction different to the set branch instruction and to the split bit instruction.

To avoid the need for state indicators, the present invention provides in another aspect a computer system for fetching, decoding and executing instructions comprising storage circuitry for holding a plurality of instructions at respective storage locations, said plurality of instructions being arranged in instruction strings, each string comprising a first instruction and a set of subsequent instructions, and instruction fetch circuitry for fetching a sequence of instructions from said storage circuitry and including an indicator for providing an indication of a next address at which a next fetch operation is to be effected. The computer system further comprises execution circuitry for executing fetched instructions, wherein at least one of said instruction strings includes a set branch instruction (SET) which provides an indication of a target location from which a subsequent instruction may be fetched, the subsequent instruction being from a different instruction string, and an effect branch instruction different from said set branch instruction and located at the branch point after which said new instructions are to be executed and wherein said instruction fetch circuitry is operated responsive to execution of a said set branch instruction (SET) to fetch in parallel subsequent instructions from said string containing said set branch instruction and new instructions from said different instruction string commencing from said target location while said subsequent instructions continue to be executed and select circuitry responsive to execution of a said effect branch (DO) instruction to cause said execution circuitry to execute said new instructions if a condition determined by the effect branch instruction is satisfied.

The invention also provides in a further aspect a method of operating a computer to fetch decode and execute instructions which computer has storage circuitry holding a plurality of instructions at respective storage locations, said plurality of instructions being arranged in instruction strings, each string comprising a first instruction and a set of subsequent instructions. The method comprises fetching a sequence of instructions from said storage circuitry and providing an indication of a next address at which a next fetch operation is to be effected, decoding said instructions, and executing each instruction in turn, wherein at least one of said instruction strings includes a set branch instruction (SET) which provides an indication of a target location from which a subsequent instruction may be fetched, the subsequent instruction being from a different instruction string. The method further comprises on execution of said set branch instruction, fetching in parallel subsequent instructions from the string containing said branch instruction and new instructions from said different instruction string commencing from said target location, continuing to execute said subsequent instructions until an effect branch instruction is executed which is located at the branch point after which new instructions are to be executed and which indicates that further instructions to be executed are said new instructions if a condition determined by the effect branch instruction is satisfied, and responding to said effect branch signal by commencing execution of said new instructions.

As an alternative arrangement to enable the computer system to perform conditional branches, it can include circuitry for holding a state indicator in one of a confirmed state and a rejected state. The confirmed state is one in which further instructions to be executed are new instructions commencing from target location. The rejected state is one in which further instructions to be executed are subsequent instructions in memory and not new instructions. One of the states can be set responsive to execution of the set branch instruction and the other of the states can be selectively set responsive to execution of a second instruction different from the set branch instruction and subject to a condition.

The second instruction can be a confirm instruction which sets the confirmed state if the confirm condition is satisfied.

Alternatively, the second instruction could be a reject instruction which sets the rejected state if the reject condition is satisfied.

The provision of these reject or confirm instructions allows a further improvement to be made in that the set branch instruction is the first instruction of the string and there is a plurality of contiguous instruction strings, with the set branch instruction acting as a further instruction to generate the effect branch signal if the state indicator is in the confirmed state. It will be appreciated that the set branch instruction acting as the further instruction will also change the state of the state indicator back to its original state. Preferably the confirm/reject instruction can be placed as early as possible within the string (after the condition has been generated) so that the execution circuitry can be given an early indication of which way the branch will go.

BRIEF DESCRIPTION OF THE DRAWINGS

For a better understanding of the present invention and to show how the same may be carried into effect, reference will now be made by way of example to the accompanying drawings.

FIG. **1** is a schematic illustrating a known branching system;

FIG. **2** is a schematic illustrating the branch system of the present invention for non-conditional branches;

FIG. **3** is a schematic illustrating the branch system of the present invention for conditional branches;

FIG. **4** is a simple block diagram of a pipelined processor;

FIG. **5** is a circuit diagram of an instruction fetcher;

FIG. **6** is a circuit diagram of a computer system for implementing branch instructions;

FIG. **7** is a circuit diagram of an instruction fetcher with kernel and descriptor functions;

FIG. **8** is a schematic diagram illustrating procedure calling;

FIG. **9** is a sketch illustrating states for performing procedure calls;

FIG. **10** is a block diagram illustrating an alternative implementation for an instruction fetch circuit;

FIG. **11** is a block diagram illustrating a non predictive fetcher; and

FIG. **12** is a block diagram illustrating a predictive fetcher.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Reference will first be made to FIGS. **2** and **3** to explain the concept underlying the branching system of the present

this agrees with the prediction, then all is well. If however the taking of the branch was incorrectly predicted, all instructions which entered the execute unit after the incorrectly predicted conditional DO instruction are deleted, and the mispredict fetcher and execute fetcher swap roles. To this end the sequencer receives a Mispredicted signal from the execution unit.

This arrangement can be implemented in a variety of different ways without affecting the concept. For example, when the fetchers are renamed the contents of the outgoing target fetcher could be copied to the incoming target fetcher. This means that the new target fetcher does not waste time refetching these instructions. It would continue the fetching from where the outgoing target fetcher left off, provided it is not full. Alternatively, the fetchers need not be renamed dynamically, but their contents transferred as appropriate. This would remove any requirement for state in the sequencer **414**.

The instruction fetcher and circuitry described above are capable of implementing so-called simple branches. Other, specialised type of branch instructions are also useful. One of these is kernel branches. Many processors have two modes of operation, one for normal programs and one for special programs. These are referred to as user and kernel modes. Kernal mode has more instructions available to it which are used to manipulate the operation of the computer. This separation is required to prevent an erroneous or malicious user mode program from causing damage to other user mode programs. Kernel mode programs can be assumed to be correct. There is therefore a need for a method to change a programs mode from user to kernel. This is done by branching to a special target location, called the kernel entry point. With the present invention this is implemented using a special set branch instruction, which does not specify the branch target location explicitly but uses an implicit value. Some state must be used to specify that when the branch occurs, the processor must change to kernel mode.

Another specialised kind of branch instruction is a so-called descriptor branch, which is a call via a pointer. This branch instruction specifies an address in memory, but it is not the address representing the target location of the branch. Instead, it is a memory location containing the target location for the branch.

FIG. **7** illustrates an instruction fetcher which can be used to implement kernel and descriptor branches. Like numerals in FIG. **7** denote like parts in FIG. **5**. The fetcher of FIG. **7** has the following additional circuitry. A kernel latch **200** holds the address to use for kernel calling and can only be programmed by trusted code. It receives at its latch input a store kernel signal **202** to latch the kernel address on line **204**. A kernel multiplexor **206** receives the kernel address at one input thereof and the normal start address on line **40** at the other input thereof. The kernel multiplexor **206** is controlled by a branch kernel signal on line **208**. The branch kernel signal is asserted at the same time as the start signals **40** would normally be asserted to initialise a branch. When the signal is asserted, the address held in the kernel latch **200** is stored into the fetch latch **65** via multiplexor **206** and a further multiplexor **208**, rather than the address supplied by the start signal.

The fetcher also includes a descriptor latch **210** which indicates whether the fetch pointer **65** holds an instruction address or a descriptor address. It is controlled by the descriptor signal on line **212**. When the descriptor latch **210** indicates that the address is a descriptor address, it is loaded from data held in the buffer **66** via a pointer register **214**. The multiplexor **208** controls whether the address supplied to the fetch pointer **65** is from the pointer register **214** or from the multiplexor **206**. A descriptor indication unit **216** signals whether or not the buffer **66** holds enough information for the pointer register **214** to hold the correct descriptor address.

The fetcher of FIG. **7** is thus able to carry out kernel branches and descriptor branches.

Another type of branch instruction is procedure calling. This requires that a suitable instruction pointer value is saved so that the procedure can return back to the piece of program from whence it came. Thus, the procedure can be called from different parts of the program. FIG. **8** illustrates the flow graph for a procedure call. FIG. **8** illustrates a program containing Part 1, Part 2 and Procedure. Part 1 has sequential blocks of instructions Block A, Block B between which is located a call instruction. Similarly, Part 2 has sequential blocks of instructions Block C, Block D between which is a call instruction. The procedure includes a sequence of procedure instructions PROC and a return instruction. The two pieces of code, Part 1 and Part 2 both call the Procedure and both return to their respective control flows. The call instruction can be implemented as a particular type of set or do branch instruction which not only identifies a target location (SET) or branch point (D0) but causes the return address of the first instruction of the, next sequential block to be saved in a return register. Then, the return instruction can be implemented as a particular type of set instruction which effects a branch to the return address which was held in the register.

FIG. **9** indicates the state register required to implement procedure calls. This includes registers **230** with a select register unit **232** controlled by a register select signal **234**. On execution of a set or do branch (or call) instruction, the address of the next instruction after the call instruction to which the program is to return is stored in the registers **230** on branch line **236** responsive to the store signal **238**. When the special set (or return) instruction is implemented, the branch is effected to the target location which is stored in the specified register **230**.

In the above described embodiment, there are two instruction fetchers which can both function as the active fetcher depending on the state of the switch multiplexor. FIG. **10** illustrate in block diagram form an alternative embodiment where the instruction fetch circuit comprises two instruction fetchers, one of which is always the active fetcher. This embodiment will now more clearly be described with reference to FIG. **10**. Like numerals in FIG. **10** denote like parts to FIG. **6**, but primed. Thus, FIG. **10** illustrates a pipelined processor **17'** including execution circuitry with a set branch instruction execution circuit **136'** and a do branch instruction execution circuit **142'**. There is an instruction pointer register **108'** and a target pointer register **118'**. The fetch circuit includes an active fetcher and a target fetcher. The active fetcher includes a fetch pointer **65'** and an instruction buffer **66'**. The target fetcher similarly includes a fetch pointer **65"** and a target instruction buffer **66"**.

On execution of a set branch instruction, the target pointer register **118'** is initialised to instruct the fetch pointer **65"** of the target fetcher to commence fetching instructions from the target location. Meanwhile, the active fetcher is fetching instructions sequentially from memory and supplying them to the processor **17'**. On execution of the effect branch instruction, a copy unit **300** acts to copy the contents of the target instruction buffer **66"** of the target fetcher to the instruction buffer **66'** of the active fetcher so that the next instructions to be supplied to the processor **17'** are those commencing from the target location.

PTO/SB/44 (04-05)
Approved for use through 04/30/2007. OMB 0651-0033
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.
(Also Form PTO-1050)

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO.	:	7047399
APPLICATION NO.	:	09/842312
ISSUE DATE	:	May 16, 2006
INVENTOR(S)	:	Andrew C. Sturges and Nathan M. Sidwell

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Col. 3, line 28, should read:
--provided a computer system for fetching, decoding and--

Col. 4, line 32 should read:
--selected one of said instruction fetchers to said execution--
line 44, should read:
--can be done in a number of ways. For example, a further--
line 49 should read:
--register can then be compared with an instruction pointer--

Col. 6, line 8 should read as shown below:
--commencing from a target location. The rejected state is one--

Col. 17, line 55 should read:
--kernel signal is asserted at the same time as the start signal--

MAILING ADDRESS OF SENDER (Please do not use customer number below):
Daniel P. McLoughlin
WOLF, GREENFIELD & SACKS, P.C.
Federal Reserve Plaza
600 Atlantic Avenue
Boston, Massachusetts 02210-2206

MAY 31 2006